Forecasting Treatment Responses Over Time Using Recurrent Marginal Structural Networks

Bryan Lim
Department of Engineering Science
University of Oxford
bryan.lim@eng.ox.ac.uk

Ahmed Alaa
Electrical Engineering Department
University of California, Los Angeles
ahmedmalaa@ucla.edu

Mihaela van der Schaar
University of Oxford
and The Alan Turing Institute
mschaar@turing.ac.uk

Abstract

Electronic health records provide a rich source of data for machine learning methods to learn dynamic treatment responses over time. However, any direct estimation is hampered by the presence of time-dependent confounding, where actions taken are dependent on time-varying variables related to the outcome of interest. Drawing inspiration from marginal structural models, a class of methods in epidemiology which use propensity weighting to adjust for time-dependent confounders, we introduce the Recurrent Marginal Structural Network - a sequence-to-sequence architecture for forecasting a patient's expected response to a series of planned treatments. Using simulations of a state-of-the-art pharmacokinetic-pharmacodynamic (PK-PD) model of tumor growth [12], we demonstrate the ability of our network to accurately learn unbiased treatment responses from observational data – even under changes in the policy of treatment assignments – and performance gains over benchmarks.

1 Introduction

With the increasing prevalence of electronic health records, there has been much interest in the use of machine learning to estimate treatment effects directly from observational data [13, 41, 44, 2]. These records, collected over time as part of regular follow-ups, provide a more cost-effective method to gather insights on the effectiveness of past treatment regimens. While the majority of previous work focuses on the effects of interventions at a single point in time, observational data also captures information on complex time-dependent treatment scenarios, such as where the efficacy of treatments changes over time (e.g. drug resistance in cancer patients [40]), or where patients receive multiple interventions administered at different points in time (e.g. joint prescriptions of chemotherapy and radiotherapy [12]). As such, the ability to accurately estimate treatment effects over time would allow doctors to determine both the treatments to prescribe and the optimal time at which to administer them.

However, straightforward estimation in observational studies is hampered by the presence of time-dependent confounders, arising in cases where interventions are contingent on biomarkers whose value are affected by past treatments. For examples, asthma rescue drugs provide short-term rapid improvements to lung function measures, but are usually prescribed to patients with reduced lung function scores. As such, naïve methods can lead to the incorrect conclusion that the medication reduces lung function scores, contrary to the actual treatment effect [26]. Furthermore, [23] show

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

that the standard adjustments for causal inference, e.g. stratification, matching and propensity scoring [16], can introduce bias into the estimation in the presence of time-dependent confounding.

Marginal structural models (MSMs) are a class of methods commonly used in epidemiology to estimate time-dependent effects of exposure while adjusting for time-dependent confounders [15, 24, 19, 14]. Using the probability of a treatment assignment, conditioned on past exposures and covariate history, MSMs typically adopt inverse probability of treatment weighting (IPTW) to correct for bias in standard regression methods [22], re-constructing a 'pseudo-population' from the observational dataset to similar to that of a randomized clinical trial. However, the effectiveness of bias correction is dependent on a correct specification of the conditional probability of treatment assignment, which is difficult to do in practice given the complexity of treatment planning. In standard MSMs, IPTWs are produced using pooled logistic regression, which makes strong assumptions on the form of the conditional probability distribution. This also requires one separate set of coefficients to be estimated per time-step and many models to be estimated for long trajectories.

In this paper, we propose a new deep learning model - which we refer to as Recurrent Marginal Structural Networks - to directly learn time-dependent treatment responses from observational data, based on in the marginal structural modeling framework. Our key contributions are as follows:

Multi-step Prediction Using Sequence-to-sequence Architecture To forecast treatment responses at multiple time horizons in the future, we propose a new RNN architecture for multi-step prediction based on sequence-to-sequence architectures in natural language processing [36]. This comprises two halves, 1) an encoder RNN which learns representations for the patient's current clinical state, and 2) a decoder which is initialized using the encoder's final memory state and computes forward predictions given the intended treatment assignments. At run time, the R-MSN also allows for prediction horizons to be flexibly adjusted to match the intended treatment duration, by expanding or contracting the number of decoder units in the sequence-to-sequence model.

Scenario Analysis for Complex Treatment Regimens Treatment planning in clinical settings is often based on the interaction of numerous variables - including 1) the desired outcomes for a patient (e.g. survival improvement or comorbidity risk reduction), 2) the treatments to assign (e.g. binary interventions or continuous dosages), and 3) the length of treatment affected by both number and duration of interventions. The R-MSN naturally encapsulates this by using multi-input/output RNNs, which can be configured to have multiples treatments and targets of different forms (e.g. continuous or discrete). Different sequences of treatments can also be evaluated using the sequence-to-sequence architecture of the network. Moreover, given the susceptibility of IPTWs to model misspecification, the R-MSN uses Long-short Term Memory units (LSTMs) to compute the probabilities required for propensity weighting. Combining these aspects together, the R-MSN is able to help clinicians evaluate the projected outcome of a complex treatment scenario – providing timely clinical decision support and helping them customize a treatment regimen to the patient. A example of scenario analysis for different cancer treatment regimens is shown in Figure 1, with the expected response of tumor growth to no treatment, chemotherapy and radiotherapy shown.

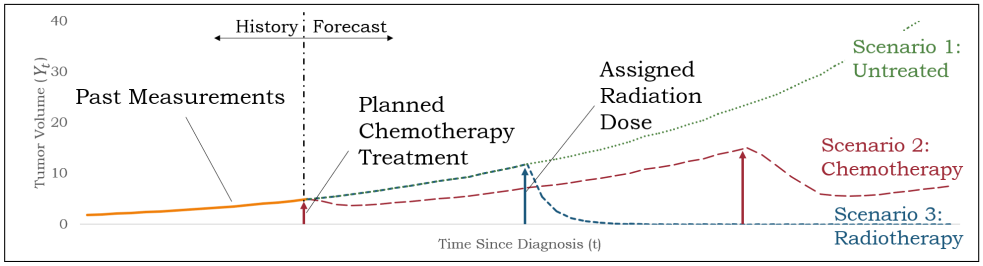


Figure 1: Forecasting Tumor Growth Under Multiple Treatment Scenarios

2 Related Works

Given the diversity of literature on causal inference, we focus on works associated with time-dependent treatment responses and deep learning here, with a wider survey in Appendix A.

G-computation and Structural Models. Counterfactual inference under time-dependent confounding has been extensively studied in the epidemiology literature, particularly in the seminal works of Robins [30, 31, 16]. Methods in this area can be categorized into 3 groups: models based on the G-computation formula, structural nested mean models, and marginal structural models [8]. While all these models provide strong theoretical foundations on the adjustments for time-dependent confounding, their prediction models are typically based on linear or logistic regression. These models would be misspecified when either the outcomes or the treatment policy exhibit complex dependencies on the covariate history.

Potential Outcomes with Longitudinal Data. Bayesian nonparametric models have been proposed to estimate the effects of both single [32, 33, 43, 34] and joint treatment assignments [35] over time. These methods use Gaussian processes (GPs) to model the baseline progression, which can estimate the treatment effects at multiple points in the future. However, some limitations do exist. Firstly, to aid in calibration, most Bayesian methods make strong assumptions on model structure - such as 1) independent baseline progression and treatment response components [43, 35], and 2) the lack of heterogeneous effects, by either omitting baseline covariates (e.g. genetic or demographic information) [34, 33] or incorporating them as linear components [43, 35]. Recurrent neural networks (RNNs) avoid the need for any explicit model specifications, with the networks learning these relationships directly from the data. Secondly, inference with Bayesian models can be computationally complex, making them difficult to scale. This arises from the use of Markov Chain-Monte Carlo sampling for g-computation, and the use of sparse GPs that have at least $O(NM^2)$ complexity, where N and M are the number of observations and inducing points respectively [39]. From this perspective, RNNs have the benefit of scalability and update their internal states with new observations as they arrive. Lastly, apart from [35] which we evaluate in Section 5, existing models do not consider treatment responses for combined interventions and multiple targets. This is handled naturally in our network by using multi-input/multi-output RNN architectures.

Deep Learning for Causal Inference. Deep learning has also been used to estimate individualized treatment effects for a single intervention at a fixed time, using instrumental variable approaches [13], generative adversarial networks [44] and multi-task architectures [3]. To the best of our knowledge, ours is the first deep learning method for time-dependent effects and establishes a framework to use existing RNN architectures for treatment response estimation.

3 Problem Definition

Let $\mathbf{Y}_{t,i} = [Y_{t,i}(1), \ldots, Y_{t,i}(\Omega_y)]$ be a vector of Ω_y observed outcomes for patient i at time t, $\mathbf{A}_{t,i} = [A_{t,i}(1), \ldots, A_{t,i}(\Omega_a)]$ a vector of actual treatment administered, $\mathbf{L}_{t,i} = [L_{t,i}(1), \ldots, L_{t,i}(\Omega_l)]$ time-dependent covariates and $\mathbf{X}_i = [X_i(1), \ldots, X_i(\Omega_v)]$ patient-specific static features. For notational simplicity, we will omit the subscript i going forward unless explicitly required.

Treatment Responses Over Time Determining an individual's response to a prescribed treatment can be characterized as learning a function $g(.)$ for the expected outcomes over a prediction horizon τ, given an intended course of treatment and past observations, i.e.:

$$\mathbb{E}\left[\mathbf{Y}_{t+\tau}|a(t,\tau-1), \bar{\mathbf{H}}_t\right] = g(\tau, a(t,\tau-1), \bar{\mathbf{H}}_t) \tag{1}$$

where $g(.)$ represents a generic, possibly non-linear, function, $a(t,\tau-1) = (\mathbf{a}_t, \ldots \mathbf{a}_{t+\tau-1})$ is an *intended* sequence of treatments $\mathbf{a}_k$ from the current time until just before the outcome is observed, and $\bar{\mathbf{H}}_t = (\bar{\mathbf{L}}_t, \bar{\mathbf{A}}_{t-1}, \mathbf{X})$ is the patient's history with covariates $\bar{\mathbf{L}}_t = (\mathbf{L}_1, \ldots, \mathbf{L}_t)$ and actions $\bar{\mathbf{A}}_{t-1} = (\mathbf{A}_1, \ldots \mathbf{A}_{t-1})$.

Inverse Probability of Treatment Weighting Inverse probability of treatment weighting, extensively studied in marginal structural modeling to adjust for time-dependent confounding [22, 16, 15, 24, 26], with extensions to joint treatment assignments [19], censored observations [14] and continuous dosages [10]. We list the key results for our problem below, with a more thorough discussion in Appendix B.

The stabilized weights for joint treatment assignments [21] can be expressed as:

$$\mathbf{SW}(t,\tau) = \prod_{n=t}^{t+\tau} \frac{f(\mathbf{A}_n|\bar{\mathbf{A}}_{n-1})}{f(\mathbf{A}_n|\bar{\mathbf{H}}_n)} = \prod_{n=t}^{t+\tau} \frac{\prod_{k=1}^{\Omega_a} f(A_n(k)|\bar{\mathbf{A}}_{n-1})}{\prod_{k=1}^{\Omega_a} f(A_n(k)|\bar{\mathbf{H}}_n)} \tag{2}$$

where $f(.)$ is the probability mass function for discrete treatment applications, or the probability density function when continuous dosages are used [10]. We also note that $\bar{\mathbf{H}}_n$ contains both past treatments $\bar{\mathbf{A}}_{n-1}$ and potential confounders $\bar{\mathbf{L}}_n$. To account for censoring, we used the additional stabilized weights below:

$$SW^*(t,\tau) = \prod_{n=t}^{t+\tau} \frac{f(C_n = 0|\mathcal{T} > n, \bar{\mathbf{A}}_{\mathbf{n-1}})}{f(C_n = 0|\mathcal{T} > n, \bar{\mathbf{L}}_{n-1}, \bar{\mathbf{A}}_{n-1}, \mathbf{X})} \tag{3}$$

where $C_n = 1$ denotes right censoring of the trajectory, and $\mathcal{T}$ is the time at which censoring occurs.

We also adopt the additional steps for stabilization proposed in [42], truncating stabilized weights at their 1st and 99th percentile values, and normalizing weights by their mean for a fixed prediction horizon, i.e. $\tilde{\mathbf{SW}} = \mathbf{SW}_i(t,\tau) / \left(\sum_{i=1}^{I} \sum_{t=1}^{T_i} \mathbf{SW}_i(t,\tau)/N \right)$ where I is the total number of patients, T_i is the length of the patient's trajectory and N the total number of observations. Stabilized weights are then used to weight the loss contributions of each training observation, expressed in squared-errors terms below for continuous predictions:

$$e(i,t,\tau) = \tilde{\mathbf{SW}}_i(t,\tau-1) \times \tilde{SW}_i^*(t,\tau-1) \times \|\mathbf{Y}_{t+\tau,i} - g(\tau, a(t,\tau-1), \bar{\mathbf{H}}_t)\|^2 \tag{4}$$

4 Recurrent Marginal Structural Networks

An MSM can be subdivided into two submodels, one modeling the IPTWs and the other estimating the treatment response itself. Adopting this framework, we use two sets of deep neural networks to build a Recurrent Marginal Structural Network (R-MSN) - 1) a set propensity networks to compute treatment probabilities used for IPTW, and 2) a prediction network used to determine the treatment response for a given set of planned interventions. Additional details on the algorithm can be found in Appendix E, with the source code uploaded onto GitHub[1].

4.1 Propensity Networks

From Equations 2 and 3, we can see that 4 key probability functions are required to calculate the stabilized weights. In all instances, probabilities are conditioned on the history of past observations ($\bar{\mathbf{A}}_{n-1}$ and $\bar{\mathbf{H}}_n$), making RNNs natural candidates to learn these functions.

Each probability function is parameterized with a different LSTM – collectively referred to as propensity networks – with action probabilities $f\left(\bar{\mathbf{A}}_n | \, . \right)$ generated jointly by a set of multi-target LSTMs and censoring probabilities $f\left(C_n = 0 | \, . \right)$ by single output LSTMs. This also accounts for possible correlations between treatment assignments, for instance in treatment regimens where complementary drugs are prescribed together to combat different aspects of the same disease.

The flexibility of RNN architectures also allows for the modeling of treatment assignments with different forms. In simple cases with discrete treatment assignments, a standard LSTM with a sigmoid output layer can be used for binary treatment probabilities or a softmax layer for categorical ones. More complex architectures, such as variational RNNs [6], can be used to compute probabilities when treatments map to continuous dosages. To calculate the binary probabilities in the experiments in Section 5, LSTMs were fitted with tanh state activations and sigmoid outputs.

4.2 Prediction Network

The prediction network focuses on forecasting the treatment response of a patient, with time-dependent confounding accounted for using IPTWs from the propensity networks. Although standard RNNs can be used for one-step-ahead forecasts, actual treatments plans can be considerably more complex, with varying durations and number of interventions depending on the condition of the patient. To remove any restrictions on the prediction horizon or number of planned interventions, we propose the sequence-to-sequence architecture depicted in Figure 4.2. One key difference between our model and standard sequence-to-sequence (e.g.[36]) is that the last unit of the encoder is also used in making predictions for the first time step, in addition to the decoder units at further horizons. This allows the R-MSN to use all available information in making predictions, including the covariates

[1] `https://github.com/sjblim/rmsn_nips_2018`

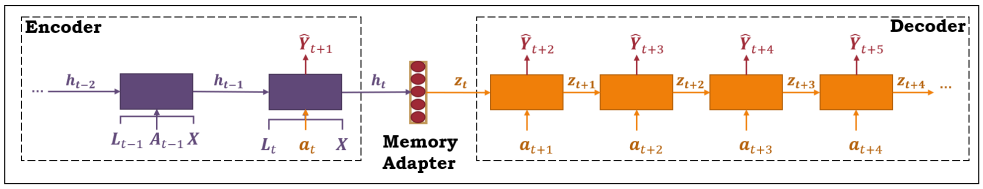


Figure 2: R-MSN Architecture for Multi-step Treatment Response Prediction

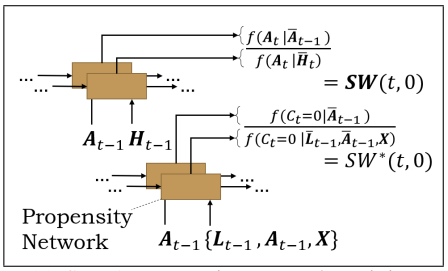
available at the current time step t. For the continuous predictions in Section 5, we used Exponential Linear Unit (ELU [7]) state activations and a linear output layer.

Encoder The goal of the encoder is to learn good representations for the patient's current clinical state, and we do so with a standard LSTM that makes one-step-ahead predictions of the outcome ($\hat{\mathbf{Y}}_{t+1}$) given observations of covariates and actual treatments. At the current follow-up time t, the encoder is also used in forecasting the expected response at $t+1$, as the latest covariate measurements L_t are available to be fed into the LSTM along with the first planned treatment assignment.

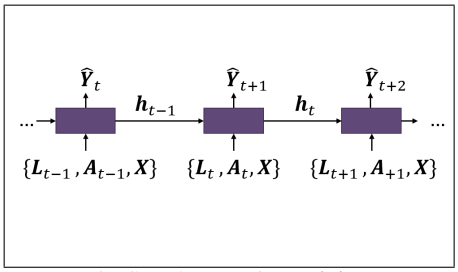
Decoder While multi-step prediction can be performed by recursively feeding outputs into the inputs at the next time step, this would require output predictions for *all* covariates, with a high degree of accuracy to reduce error propagation through the network. Given that often only a small subset treatment outcomes are of interest, it would be desirable to forecast treatment responses on the basis of planned future actions alone. As such, the purpose of the decoder is to propagate the encoder representation forwards in time - using only the proposed treatment assignments and avoiding the need to forecast input covariates. This is achieved by training another LSTM that accepts only actions as inputs, but initializing the internal memory state of the first LSTM in the decoder sequence (z_t) using encoder representations. To allow for different state sizes in the encoder and decoder, encoder internal states (h_t) are passed through a single network layer with ELU activations, i.e. the memory adapter, before being initializing the decoder. As the network is made up of LSTM units, the internal states here refer to the concatenation of the cell and hidden states [17] of the LSTM.

4.3 Training Procedure

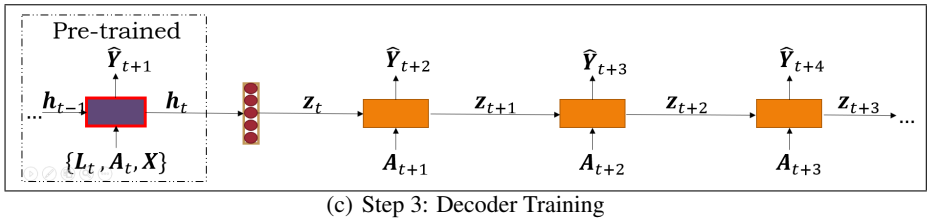
The training procedure for R-MSNs can be subdivided into the 3 training steps shown in Figure 3 - starting with the propensity networks, followed by the encoder, and ending with the decoder.



(a) Step 1: Propensity Network Training



(b) Step 2: Encoder Training



(c) Step 3: Decoder Training

Figure 3: Training Procedure for R-MSNs

Step 1: Propensity Network Training From Figure 3(a), each propensity network is first trained to estimate the probability of the treatment assigned at each time step, which is combined to compute $\mathbf{SW}(t,0)$ and $SW^*(t,0)$ at each time step. Stabilized weights for longer horizons can then be obtained from their cumulative product, i.e. $\mathbf{SW}(t,\tau) = \prod_{j=0}^{\tau} \mathbf{SW}(t+j,0)$. For tests in Section 5, propensity networks were trained using standard binary cross entropy loss, with treatment assignments and censoring treated as binary observations.

Step 2: Encoder Training Next, decoder and encoder training was divided into separate steps - accelerating learning by first training the encoder to learn representations of the patient's clinical state and then using the decoder to extrapolate them according to the intended treatment plan. As such, the encoder was trained to forecast standard one-step-ahead treatment response according to the structure in Figure 3(b), using all available information on treatments and covariates until the current time step. Upon completion, the encoder was used to perform a feed-forward pass over the training and validation data, extracting the internal states $\mathbf{h}_t$ for the final training step. As tests in Section 5 were performed for continuous outcomes, we express the loss function for the encoder as a weighted mean-squared error loss ($\mathcal{L}_{encoder}$ in Equation 5), although we note that this approach is compatible with other loss functions, e.g. cross entropy for discrete outcomes.

Step 3: Decoder Training Finally, the decoder and memory adapter were trained together based on the format in Figure 3(c). For a given patient, observations were batched into shorter sequences of up to τ_{max} steps, such that each sequence commencing at time t is made up of $[\mathbf{h}_t, \{\mathbf{A}_{t+1}, \ldots, \mathbf{A}_{t+\tau_{max}-1}\}, \{\mathbf{Y}_{t+2}, \ldots, \mathbf{Y}_{t+\tau_{max}}\}]$. These were compiled for all patient-times and randomly grouped into minibatches to be used for backpropagation through time. For continuous predictions, the loss function for the decoder is ($\mathcal{L}_{decoder}$) can also be found in Equation 5.

$$\mathcal{L}_{encoder} = \sum_{i=1}^{I} \sum_{t=1}^{T_i} e(i,t,1) \qquad \mathcal{L}_{decoder} = \sum_{i=1}^{I} \sum_{t=1}^{T_i} \sum_{\tau=2}^{min(T_i-t,\tau_{max})} e(i,t,\tau) \tag{5}$$

5 Experiments With Cancer Growth Simulation Model

5.1 Simulation Details

As confounding effects in real-world datasets are unknown a priori, methods for treatment response estimation are often evaluated using data simulations, where treatment application policies are explicitly modeled [34, 33, 35]. To ensure that our tests are fully reproducible and realistic from a medical perspective, we adopt the pharmacokinetic-pharmacodynamic (PK-PD) model of [12] - the state-of-the-art in treatment response modeling for non-small cell lung patients. The model features key characteristics present in actual lung cancer treatments, such as combined effects of chemo- and radiotherapy, cell repopulation after treatment, death/recovery of patients, and different staring distributions of tumor sizes based on the stage of cancer at diagnosis. On the whole, PK-PD models allow clinicians to explore hypotheses around dose-response relationships and propose optimal treatment schedules [5, 29, 11, 9, 1]. While we refer readers to [12] for the finer details of the model, such as specific priors used, we examine the overall structure of the model below to illustrate treatment-response relationships and how time-dependent confounding is introduced.

PK-PD Model for Tumor Dynamics We use a discrete-time model for tumor volume $V(t)$, where t is the number of days since diagnosis:

$$V(t) = \Bigg(1 + \underbrace{\rho \log(\frac{K}{V(t-1)})}_{\text{Tumor Growth}} - \underbrace{\beta_c C(t)}_{\text{Chemotherapy}} - \underbrace{(\alpha d(t) + \beta d(t)^2)}_{\text{Radiation}} + \underbrace{e_t}_{\text{Noise}}\Bigg) V(t-1) \tag{6}$$

where ρ, K, β_c, α, β are model parameters sampled for each patient according to prior distributions in [12]. A Gaussian noise term $e_t \sim N(0, 0.01^2)$ was added to account for randomness in the growth of the tumor. $d(t)$ is the dose of radiation applied at t, while drug concentration C(t) is modeled according to an exponential decay with a half life of 1 day, i.e.:

$$C(t) = \tilde{C}(t) + C(t-1)/2 \tag{7}$$

where $\tilde{C}(t)$ is an new continuous dose of chemotherapy drugs applied at time t. To account for heterogeneous effects, we added static features to the simulation model by randomly subclassing

patients into 3 different groups, with each patient having a group label $S_i \in \{1, 2, 3\}$. This represents specific characteristics which affect with patient's response to chemotherapy and radiotherapy (e.g. by genetic factors [4]), which augment the prior means of β_c and α according to:

$$\mu'_{\beta_c}(i) = \begin{cases} 1.1\mu_{\beta_c} \text{ , if } S_i = 3 \\ \mu_{\beta_c} \text{ , otherwise} \end{cases} \qquad \mu'_{\alpha}(i) = \begin{cases} 1.1\mu_{\alpha} \text{ , if } S_i = 1 \\ \mu_{\alpha} \text{ , otherwise} \end{cases} \tag{8}$$

where μ_* are the mean parameters of [12], and $\mu'_*(i)$ those used to simulate patient i. We note that the value of β is set in relation to α, i.e. $\alpha/\beta = 10$, and would also be adjusted accordingly by S_i.

Censoring Mechanisms Patient censoring is incorporated by modeling 1) death when tumor diameters reach $D_{max} = 13\ cm$ (or a volume of $V_{max} = 1150\ cm^3$ assuming perfectly spherical tumors), 2) recovery determined by a Bernoulli process with recovery probability $p_t = \exp(-V_t)$, and 3) termination of observations after 60 days (administrative censoring).

Treatment Assignment Policy To introduce time-dependent confounders, we assume that chemotherapy prescriptions $A_c(t) \in \{0, 1\}$ and radiotherapy prescriptions $A_d(t) \in \{0, 1\}$ are Bernoulli random variables, with probabilities $p_c(t)$ and $p_d(t)$ respectively that are a functions of the tumor diameter:

$$p_c(t) = \sigma\left(\frac{\gamma_c}{D_{max}}(\bar{D}(t) - \theta_c)\right) \qquad p_d(t) = \sigma\left(\frac{\gamma_d}{D_{max}}(\bar{D}(t) - \theta_d)\right) \tag{9}$$

where $\bar{D}(t)$ is the average tumor diameter over the last 15 days, $\sigma(.)$ is the sigmoid activation function, and θ_* and γ_* are constant parameters. θ_* is fixed such that $\theta_c = \theta_d = D_{max}/2$, giving the model a 0.5 probability of treatment application exists when the tumor is half its maximum size. When treatments are applied, i.e. $A_c(t)$ or $A_d(t)$ is 1, chemotherapy is assumed to be administered in $5.0\ mg/m^3$ doses of Vinblastine, and radiotherapy in $2.0\ Gy$ fractions. γ also controls the degree of time-dependent confounding - starting with no confounding at $\gamma = 0$, as treatment assignments are independent of the response variable, and an increase as γ becomes larger.

5.2 Benchmarks

We evaluate the performance of R-MSNs against MSMs and Bayesian nonparametric models, focusing on its effectiveness in estimating unbiased treatment responses and its multi-step prediction performance. An overview of the models tested is summarized below:

Standard Marginal Structural Models (MSM) For the MSMs used in our investigations, we adopt similar approximations to [19, 14], encoding historical actions via cumulative sum of applied treatments, e.g. $cum(\bar{a}_c(t-1)) = \sum_{k=1}^{t-1} a_c(k)$, and covariate history using the previous observed value $V(t-1)$. The exact forms of the propensity and prediction models are in Appendix D.

Bayesian Treatment Response Curves (BTRC) We also benchmark our performance against the model of [35] - the state-of-the-art in forecasting multistep treatment responses for joint therapies with multiple outcomes. Given that the simulation model only has one target outcome, we also consider a simpler variant of the model without "shared" components, denoting this as the reduced BTRC (R-BTRC) model. This reduced parametrization was found to improve convergence during training, and additional details on calibration can be found in Appendix G.

Recurrent Marginal Structural Networks (R-MSN) R-MSNs were designed according to the description in Section 4, with full details on training and hyperparameter in Appendix F. To evaluate the effectiveness of the propensity networks, we also trained predictions networks using the IPTWs from the MSM, including this as an additional benchmark in Section 5.3 (Seq2Seq + Logistic).

5.3 Performance Evaluations

Time-Dependent Confounding Adjustments To investigate how well models learn unbiased treatment responses from observational data, we trained all models on simulations with $\gamma_c = \gamma_d = 10$ (biased policy) and examine the root-mean-squared errors (RMSEs) of one-step-ahead predictions as γ_* is reduced. Both γ_* parameters were set to be equal in this section for simplicity, i.e. $\gamma_c = \gamma_d = \gamma$. Using the simulation model in Section 5.1, we simulated 10,000 paths to be used for model training, 1,000 for validation data used in hyperparameter optimization, and another 1,000 for out-of-sample

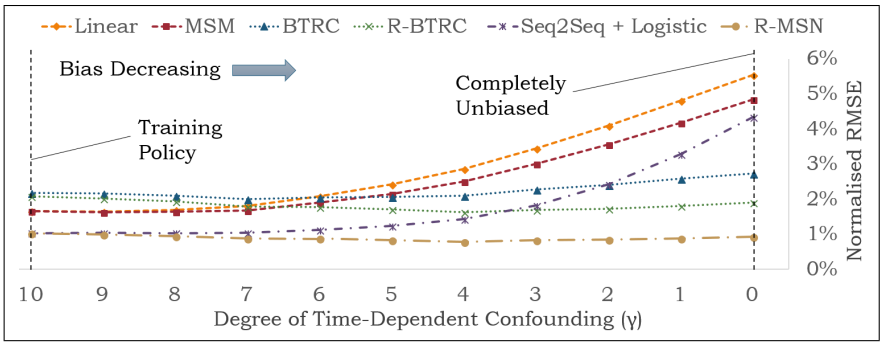


Figure 4: Normalized RMSEs for One-Step-Ahead Predictions

testing. For linear and MSM models, which do not have hyperparameters to optimized, we combined both training and validation datasets for model calibration.

Figure 4 shows the RMSE values of various models at different values of γ, with RMSEs normalized with V_{max} and reported in percentage terms. Here, we focus on the main comparisons of interest – 1) linear models to provide a baseline on performance, 2) linear vs MSMs to evaluate traditional methods for IPTWs, 3) Seq2Seq + logistic IPTWs vs MSMs for the benefits of the Seq2Seq model, 4) R-MSN vs Seq2Seq + logistic to determine the improvements of our model and RNN-estimated IPTWs, and 5) BTRC/R-BTRC to benchmark against state-of-the-art methods. Additional results are also documented in Appendix C for reference.

From the graph, R-MSNs displayed the lowest RMSEs across all values of γ, decreasing slightly from a normalized RMSE of 1.02% at $\gamma = 10$ to 0.92% at $\gamma = 0$. Focusing on RMSEs at $\gamma = 0$, R-MSNs improve MSMs by 80.9% and R-BTCs by 66.1%, demonstrating its effectiveness in learning unbiased treatment responses from confounded data. The propensity networks also improve unbiased treatment estimates by 78.7% (R-MSN vs. Seq2Seq + Logistic), indicating the benefits of more flexible models for IPTW estimation. While the IPTWs of MSMs do provide small gains for linear models, linear models still exhibit the largest unbiased RMSE across all benchmarks - highlighting the limitations of linear models in estimating complex treatment responses. Bayesian models also perform consistently across γ, with normalized RMSEs for R-BTRC decreasing from 2.09% to 1.91% across $\gamma = 0$ to 10, but were also observed to slightly underperform linear models on the training data itself. Part of this can potentially be attributed to model misspecification in the BTRC, which assumes that treatment responses are linear time-invariant and independent of the baseline progression. The differences in modeling assumptions can be seen from Equation 6, where chemotherapy and radiotherapy contributions are modeled as multiplicative with $V(t)$. This highlights the benefits of the data-driven nature of the R-MSN, which can flexibly learn treatment response models of different types.

Multi-step Prediction Performance To evaluate the benefits of the sequence-to-sequence architecture, we report the normalized RMSEs for multi-step prediction in Table 1, using the best model of each category (R-MSN, MSM and R-BTRC). Once again, the R-MSN outperforms benchmarks for all timesteps, beating MSMs by 61% on the training policy and 95% for the unbiased one. While the R-BTRC does show improvements over MSMs for the unbiased treatment response, we also observe a slight underperformance versus MSMs on the training policy itself, highlighting the advantages of R-MSNs.

6 Conclusions

This paper introduces Recurrent Marginal Structural Networks - a novel learning approach for predicting unbiased treatment responses over time, grounded in the framework of marginal structural models. Networks are subdivided into two parts, a set of propensity networks to accurately compute the IPTWs, and a sequence-to-sequence architecture to predict responses using only a planned sequence of future actions. Using tests on a medically realistic simulation model, the R-MSN demonstrated performance improvements over traditional methods in epidemiology and the state-of-the-art models for joint treatment response prediction over multiple timesteps.

Table 1: Normalized RMSE for Various Prediction Horizons τ

	τ	**1**	**2**	**3**	**4**	**5**	**Ave. % Decrease in RMSE vs MSMs**
Training Policy ($\gamma_c = 10, \gamma_d = 10$)	MSM	1.67%	2.51%	3.12%	3.64%	4.09%	-
	R-BTRC	2.09%	2.85%	3.50%	4.07%	4.58%	-32% (↑ RMSE)
	R-MSN	**1.02%**	**1.80%**	**1.90%**	**2.11%**	**2.46%**	**+61%**
Unbiased Assignment ($\gamma_c = 0, \gamma_d = 0$)	MSM	4.84%	5.29%	5.51%	5.65%	5.84%	-
	R-BTRC	1.91%	2.74%	3.34%	3.75%	4.08%	+66%
	R-MSN	**0.92%**	**1.38%**	**1.30%**	**1.22%**	**1.14%**	**+95%**
Unbiased Radiotherapy ($\gamma_c = 10, \gamma_d = 0$)	MSM	3.85%	4.03%	4.32%	4.60%	4.91%	-
	R-BTRC	1.74%	1.68%	2.14%	2.54%	2.91%	+74%
	R-MSN	**1.08%**	**1.66%**	**1.83%**	**1.98%**	**2.14%**	**+84%**
Unbiased Chemotherapy ($\gamma_c = 0, \gamma_d = 10$)	MSM	1.84%	2.65%	3.09%	3.44%	3.83%	-
	R-BTRC	1.16%	2.45%	2.97%	3.34%	3.64%	+20%
	R-MSN	**0.65%**	**1.13%**	**1.05%**	**1.17%**	**1.31%**	**+87%**

Acknowledgments

This research was supported by the Oxford-Man Institute of Quantitative Finance, the US Office of Naval Research (ONR), and the Alan Turing Institute.

References

[1] Optimizing drug regimens in cancer chemotherapy: a simulation study using a pk–pd model. *Computers in Biology and Medicine*, 31(3):157 – 172, 2001. Goal-Oriented Model-Based drug Regimens.

[2] Ahmed M. Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Proceedings of the thirty-first Conference on Neural Information Processing Systems, (NIPS)*, 2017.

[3] Ahmed M. Alaa, Michael Weisz, and Mihaela van der Schaar. Deep counterfactual networks with propensity dropout. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[4] H. Bartsch, H. Dally, O. Popanda, A. Risch, and P. Schmezer. Genetic risk profiles for cancer susceptibility and therapy response. *Recent Results Cancer Res.*, 174:19–36, 2007.

[5] Letizia Carrara, Silvia Maria Lavezzi, Elisa Borella, Giuseppe De Nicolao, Paolo Magni, and Italo Poggesi. Current mathematical models for cancer drug discovery. *Expert Opinion on Drug Discovery*, 12(8):785–799, 2017.

[6] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2980–2988, Cambridge, MA, USA, 2015.

[7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *CoRR*, abs/1511.07289, 2015.

[8] R.M. Daniel, S.N. Cousens, B.L. De Stavola, M. G. Kenward, and J. A. C. Sterne. Methods for dealing with time-dependent confounding. *Statistics in Medicine*, 32(9):1584–1618, 2012.

[9] Mould DR, Walz A-C, Lave T, Gibbs JP, and Frame B. Developing exposure/response models for anticancer drug treatment: Special considerations. *CPT: Pharmacometrics & Systems Pharmacology*, 4(1):12–27.

[10] Peter H. Egger and Maximilian von Ehrlich. Generalized propensity scores for multiple continuous treatment variables. *Economics Letters*, 119(1):32 – 34, 2013.

[11] M. J. Eigenmann, N. Frances, T. Lavé, and A.-C. Walz. Pkpd modeling of acquired resistance to anti-cancer drug treatment. *Journal of Pharmacokinetics and Pharmacodynamics*, 44(6):617–630, 2017.

[12] Changran Geng, Harald Paganetti, and Clemens Grassberger. Prediction of treatment response for combined chemo- and radiation therapy for non-small cell lung cancer patients using a bio-mathematical model. *Scientific Reports*, 7, 2017.

[13] Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. Deep IV: A flexible approach for counterfactual prediction. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[14] Miguel A. Hernan, Babette Brumback, and James M. Robins. Marginal structural models to estimate the joint causal effect of nonrandomized treatments. *Journal of the American Statistical Association*, 96(454):440–448, 2001.

[15] Miguel A. Hernan and James M. Robins. Marginal structural models to estimate the causal effect of zidovudine on the survival of hiv-positive men. *Epidemiology*, 359:561–570, 2000.

[16] MA Hernán and JM Robins. *Causal Inference*. Chapman & Hall/CRC, 2018.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[18] William Hoiles and Mihaela van der Schaar. A non-parametric learning method for confidently estimating patient's clinical state and dynamics. In *Proceedings of the twenty-ninth Conference on Neural Information Processing Systems, (NIPS)*, 2016.

[19] Chanelle J. Howe, Stephen R. Cole, Shruti H. Mehta, and Gregory D. Kirk. Estimating the effects of multiple time-varying exposures using joint marginal structural models: alcohol consumption, injection drug use, and hiv acquisition. *Epidemiology*, 23(4):574–582, 2012.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[21] Clovis Lusivika-Nzinga, Hana Selinger-Leneman, Sophie Grabar, Dominique Costagliola, and Fabrice Carrat. Performance of the marginal structural cox model for estimating individual and joined effects of treatments given in combination. *BMC Medical Research Methodology*, 17(1):160, Dec 2017.

[22] Mohammad Ali Mansournia, Mahyar Etminan, Goodarz Danaei, Jay S Kaufman, and Gary Collins. A primer on inverse probability of treatment weighting and marginal structural models. *Emerging Adulthood*, 4(1):40–59, 2016.

[23] Mohammad Ali Mansournia, Mahyar Etminan, Goodarz Danaei, Jay S Kaufman, and Gary Collins. Handling time varying confounding in observational research. *BMJ*, 359, 2017.

[24] Mohammad Alia Mansournia, Goodarzc Danaei, Mohammad Hosseind Forouzanfar, Mahmoodb Mahmoodi, Mohsene Jamali, Nasrinf Mansournia, and Kazemb Mohammad. Effect of physical activity on functional performance and knee pain in patients with osteoarthritis : Analysis with marginal structural models. *Epidemiology*, 23(4):631–640, 2012.

[25] Charles E McCulloch and Shayle R. Searle. *Generalized, Linear and Mixed Models*. Wiley, New York, 2001.

[26] Kathleen M. Mortimer, Romain Neugebauer, Mark van der Laan, and Ira B. Tager. An application of model-fitting procedures for marginal structural models. *American Journal of Epidemiology*, 162(4):382–388, 2005.

[27] Mihaela van der Schaar Onur Atan, James Jordon. Deep-treat: Learning optimal personalized treatments from observational data using neural networks. In *AAAI*, 2018.

[28] Mihaela van der Schaar Onur Atan, William Zame. Constructing effective personalized policies using counterfactual inference from biased data sets with many features. In *Machine Learning*, 2018.

[29] Kyungsoo Park. A review of modeling approaches to predict drug response in clinical oncology. *Yonsei Medical Journal*, 58(1):1–8, 2017.

[30] Thomas S. Richardson and Andrea Rotnitzky. Causal etiology of the research of james m. robins. *Statistical Science*, 29(4):459–484, 2014.

[31] James M. Robins, Miguel Ángel Hernán, and Babette Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):550–560, 2000.

[32] Jason Roy, Kirsten J. Lum, and Michael J. Daniels. A bayesian nonparametric approach to marginal structural models for point treatments and a continuous or survival outcome. *Biostatistics*, 18(1):32–47, 2017.

[33] Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. In *Proceedings of the thirty-first Conference on Neural Information Processing Systems, (NIPS)*, 2017.

[34] Ricardo Silva. Observational-interventional priors for dose-response learning. In *Proceedings of the Thirtieth Conference on Neural Information Processing Systems, (NIPS)*, 2016.

[35] Hossein Soleimani, Adarsh Subbaswamy, and Suchi Saria. Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[36] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the twenty-seventh Conference on Neural Information Processing Systems, (NIPS)*, 2014.

[37] Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16:1731–1755, 2015.

[38] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. *CoRR*, abs/1502.02362, 2015.

[39] Michalis K. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the twelth International Conference on Artificial Intelligence and Statistic, (AISTATS)*, 2009.

[40] Panagiotis J. Vlachostergios and Bishoy M. Faltas. Treatment resistance in urothelial carcinoma: an evolutionary perspective. *Nature Review Clincal Oncology*, 2018.

[41] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 2017.

[42] Yongling Xiao, Michal Abrahamowicz, and Erica Moodie. Accuracy of conventional and marginal structural cox model estimators: A simulation study. *The International Journal of Biostatistics*, 6(2), 2010.

[43] Yanbo Xu, Yanxun Xu, and Suchi Saria. A non-parametric bayesian approach for estimating treatment-response curves from sparse time series. In *Proceedings of the 1st Machine Learning for Healthcare Conference (MLHC)*, 2016.

[44] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations (ICLR)*, 2018.

Appendix

A Extended Related Works

Potential Outcomes with Cross-sectional Data. A simpler instantiation of the problem is to estimate the effect of a treatment applied to subjects in a (static) cross-sectional dataset. This problem has recently attracted a lot of attention in the machine learning community, and various interesting ideas were proposed to account for selection bias [3, 41, 44]. Unfortunately, most of these works cast the treatment effect estimation problem as one of learning under "covariate shift", where the goal is to learn a model for the outcomes that generalizes well to a population were treatments are randomly assigned to the subjects. Because of the sequential nature of the treatment assignment process in our setup, estimating treatment effects under time-dependent confounding cannot be similarly framed as a covariate shift problem, and hence the ideas developed in those works cannot be straightforwardly applied to our setup.

Off-policy Evaluation. A closely related problem in the area of reinforcement learning is the problem of off-policy evaluation using retrospective observational data, also known as "logged bandit feedback" [38, 37, 18, 27, 28]. In this problem, the goal is to use sequences of states, actions and rewards generated by a decision-maker that operates under an unknown policy in order to estimate the expected reward of a given policy. In our setting, we focus on estimating a trajectory of outcomes given an application of a treatment (or a sequence of treatments) rather than estimating the average reward of a policy, and hence the "counterfactual risk minimization" framework in [37] would not result in optimal estimates in our setup. However, our learning model – with a different objective function– can be applied for the problem of off-policy evaluation.

B Background on Marginal Structural Models

In this section, we summarize the key relevant points from the seminal paper of Robins [31]. Without loss of generality, we consider the case of univariate treatments, response variables and baseline covariates here for simplicity.

Marginal structural models are typically considered in the context of follow-up studies, for example in patients with HIV [31]. Time in the study is typically measured in relation to a fixed starting point, such as the first follow-up date or time of diagnosis (i.e. $t = 1$). In such settings, marginal structural models are used to measure the average treatment effect conditioned on a series of potential actions and baseline covariate V taken at the start of the study, expressed in the form:

$$\mathbb{E}\left[Y_\tau | a_1, \ldots, a_\tau, V\right] = r(a_1, \ldots, a_\tau, X; \Theta) \tag{10}$$

where $r(.)$ is a generic, typically linear, function with parameters Θ.

Time-Dependent Confounding. A full description of time-varying confounding can be found in [23], with formal definitions in [14]. Time-dependent confounding in observational studies arises as confounders have values which change over time - for example in cases where treatments are moderated based on the patient's response. A causal graph for 2-step study can be found in Figure 5, where U denotes unmeasured factors. Note that U_0, U_1 do not have arrows to actions assignments, reflecting the assumption of no unmeasured confounding.

Inverse Probability of Treatment Weighting From [31], under assumptions of no unmeasured confounding, positivity, and correct model specification, the stabilized IPTWs can be expressed as:

$$SW(\tau) = \prod_{n=0}^{\tau} \frac{f(A_n | \bar{A}_{n-1})}{f(A_n | \bar{A}_{n-1}, \bar{L}_n, X)} \tag{11}$$

Noting that V is defined to be a subset of L_0 in [31]. Informally, they note the denominator to be conditional probability of a treatment assignment given past observations of treatment assignments and covariates and the numerator being that of treatment assignments alone, with the stabilized weights representing the incremental adjustment between the two.

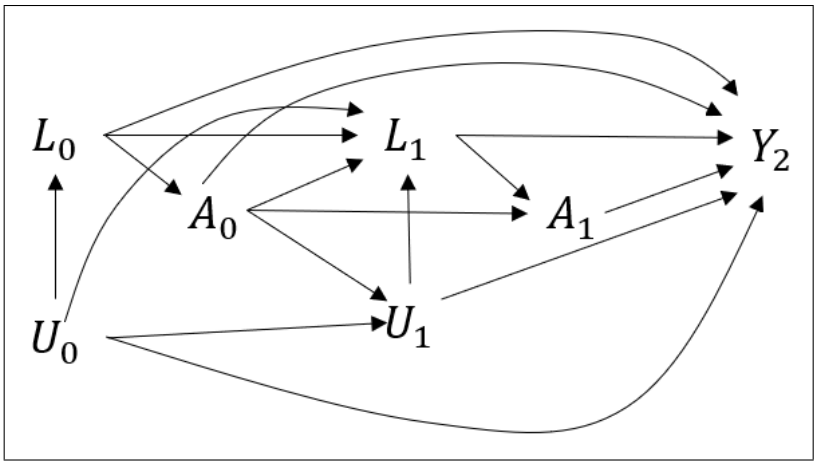


Figure 5: Causal Graph of Time-dependent Confounding for 2-step Study

In real clinical settings, it is often desirable to determine the treatment response in relation to the current follow up time, given past information. As such, we consider trajectories in relation to the last follow-up time t, retaining the form of the stabilized weights of the MSM and using all past observations, i.e.

$$SW(t,\tau) = \prod_{n=t}^{t+\tau} \frac{f(A_n|\bar{A}_{n-1})}{f(A_n|\bar{A}_{n-1},\bar{L}_n,X)} \tag{12}$$

C Additional Results for Experiments with Cancer Growth Simulation

Table 2 documents the full list of comparison for one-step-ahead predictions when tested for various γ, using different combinations of prediction and IPTW models.

$\gamma =$	**0**	**1**	**2**	**3**	**4**	**5**
Linear (No IPTWs)	5.55%	4.81%	4.09%	3.44%	2.86%	2.42%
MSM	4.84%	4.19%	3.56%	3.00%	2.51%	2.15%
MSM (LSTM IPTWs)	4.26%	3.68%	3.13%	2.64%	2.22%	1.95%
Seq2Seq (No IPTWs)	1.52%	1.39%	1.28%	1.23%	1.17%	1.17%
Seq2Seq (Logistic IPTWs)	4.34%	3.28%	2.42%	1.83%	1.43%	1.23%
R-MSN	0.92%	0.89%	0.85%	0.84%	0.79%	0.84%
BTRC	2.73%	2.59%	2.42%	2.28%	2.11%	2.07%
R-BTRC	1.91%	1.81%	1.72%	1.69%	1.63%	1.71%
$\gamma =$	**6**	**7**	**8**	**9**	**10**	
Linear (No IPTWs)	2.09%	1.80%	1.70%	1.65%	1.66%	
MSM	1.90%	1.68%	1.64%	1.64%	1.67%	
MSM (LSTM IPTWs)	1.77%	1.61%	1.62%	1.64%	1.69%	
Seq2Seq (No IPTWs)	1.13%	1.07%	1.07%	1.09%	1.08%	
Seq2Seq (Logistic IPTW)	1.12%	1.04%	1.04%	1.05%	1.04%	
R-MSN	0.88%	0.88%	0.94%	1.00%	1.02%	
BTRC	2.05%	2.01%	2.10%	2.16%	2.19%	
R-BTRC	1.77%	1.79%	1.93%	2.02%	2.09%	

Table 2: One-step-ahead Prediction Performance for Models Calibrated on $\gamma = 10$

D Marginal Structural Models for Cancer Simulation

The probabilities required for the IPTWs of the standard MSM in Section 5.2 can be described using logistic regression models with equations below:

$$f(A_t(k)|\bar{\mathbf{A}}_t) = \sigma\big(\omega_1^{(k)}\,(\sum_{n=0}^{t}\bar{A}_c(n-1)) + \omega_2^{(k)}\,(\sum_{n=0}^{t}\bar{A}_d(n-1))\big) \tag{13}$$

$$\begin{aligned} f(A_t(k)|\bar{\mathbf{H}}_t) = \sigma\big(\omega_5^{(k)}\,(\sum_{n=0}^{t}\bar{A}_c(n-1)) + \omega_6^{(k)}\,\sum_{n=0}^{t}(\bar{A}_d(n-1)) \\ +\omega_7^{(k)}\,V(t) + \omega_8^{(k)}\,V(t-1) + \omega_9^{(k)}\,S\big) \end{aligned} \tag{14}$$

$$f(C_t = 0|\mathcal{T} > n, \bar{\mathbf{A}}_{n-1}) = \sigma\big(\omega_{10}\,(\sum_{n=0}^{t}\bar{A}_c(n-1)) + \omega_{11}\,\sum_{n=0}^{t}(\bar{A}_d(n-1))\big) \tag{15}$$

$$\begin{aligned} f(C_t = 0|\mathcal{T} > n, \bar{\mathbf{L}}_{n-1}, \bar{\mathbf{A}}_{n-1}, \mathbf{X}) = \sigma\big(\omega_{12}\,(\sum_{n=0}^{t}\bar{A}_c(n-1)) + \omega_{13}\,(\sum_{n=0}^{t}\bar{A}_d(n-1)) \\ +\omega_{14}\,V(t-1) + \omega_{15}\,S\big) \end{aligned} \tag{16}$$

where $\sigma(.)$ the sigmoid function and ω_* are regression coefficients.

The regression model for prediction is given by:

$$\begin{aligned} g(\tau, a(t,\tau-1), \bar{\mathbf{H}}_t) = \beta_1\,(\sum_{n=0}^{t}\bar{A}_c(n-1)) + \beta_2\,(\sum_{n=0}^{t}\bar{A}_d(n-1)) \\ +\beta_3\,V(t) + \beta_4\,V(t-1) + \beta_5\,S \end{aligned} \tag{17}$$

E Algorithm Description for R-MSNs

To provide additional clarity on the relationship between the propensity networks and the Seq2Seq model, the pseudocode in Algorithm 1 describes the training process mentioned in Section 4.3.

We first define function $r_{(.)}\left(.;\boldsymbol{\theta}_{(.)}\right)$ to be RNN outputs given a vector of weights and hyperparameters $\boldsymbol{\theta}_{(.)}$. We refer the reader to Section 3 for more information on the functions in the MSM framework approximated by RNNs.

Propensity Networks Components of the propensity networks are used to compute the IPTWs $\mathbf{SW}(t,\tau)$ and $SW^*(t,\tau)$ as defined in Equations 2 and 3 respectively. The probabilities in the numerators and denominators are taken to be outputs of the propensity networks as below:

$$f(\mathbf{A}_n|\bar{\mathbf{A}}_{n-1}) = r_{A1}(\mathbf{A}_n|\bar{\mathbf{A}}_{n-1};\boldsymbol{\theta}_{A1}) \tag{18}$$

$$f(\mathbf{A}_n|\bar{\mathbf{H}}_n) = r_{A2}(\mathbf{A}_n|\bar{\mathbf{H}}_n;\boldsymbol{\theta}_{A2}) \tag{19}$$

$$f(C_n = 0|\mathcal{T} > n, \bar{\mathbf{A}}_{\mathbf{n-1}}) = r_{C1}(\bar{\mathbf{A}}_{\mathbf{n-1}};\boldsymbol{\theta}_{C1}) \tag{20}$$

$$f(C_n = 0|\mathcal{T} > n, \bar{\mathbf{L}}_{n-1}, \bar{\mathbf{A}}_{n-1}, \mathbf{X}) = r_{C1}(\bar{\mathbf{L}}_{n-1}, \bar{\mathbf{A}}_{n-1}, \mathbf{X};\boldsymbol{\theta}_{C2}) \tag{21}$$

Encoder The encoder is also defined in a similar fashion below, with an additional function to output the internal states of the LSTM $\tilde{g}_{E1}(\bar{\mathbf{L}}_t, \bar{\mathbf{A}}_t, \mathbf{X};\boldsymbol{\theta}_{E1})$. The encoder also computes the one-step-ahead predictions, i.e. $g(1, a(t,0), \bar{\mathbf{H}}_t)$ as per Equation 1, which is to define the prediction error $e(i,t,1)$ and encoder loss $\mathcal{L}_{encoder}$ – i.e.Equations 4 and 5 respectively.

$$g(1, a(t,0), \bar{\mathbf{H}}_t) = r_E(\bar{\mathbf{L}}_t, \bar{\mathbf{A}}_t, \mathbf{X};\boldsymbol{\theta}_E) \tag{22}$$

$$\mathbf{h}_t = \tilde{r}_E(\bar{\mathbf{L}}_t, \bar{\mathbf{A}}_t, \mathbf{X};\boldsymbol{\theta}_E) \tag{23}$$

Decoder The decoder then uses the seq2seq architecture to project encoder states $\mathbf{h}_t$ forwards in time, incorporating planned future actions $\mathbf{a}_{t+\tau}$. This is also combined with the IPTWs to define the decoder loss $\mathcal{L}_{decoder}$ in Equation 5.

$$g(\tau, a(t, \tau-1), \bar{\mathbf{H}}_t) = r_D(\mathbf{h}_t, a_{t+1}, \ldots, a_{t+\tau}; \boldsymbol{\theta}_D), \forall \tau > 1 \tag{24}$$

Algorithm 1 Training Process for R-MSN

Input: Training/Validation Data $\bar{\mathbf{L}}_{1:T},\ \bar{\mathbf{A}}_{1:T},\ X$
Output: Neural network weights and hyperparameters for:
1) $\mathbf{SW}(t,\tau)$ networks: $\boldsymbol{\theta}_{A1},\ \boldsymbol{\theta}_{A2}$
2) $SW^*(t,\tau)$ networks: $\boldsymbol{\theta}_{C1},\ \boldsymbol{\theta}_{C2}$
3) Encoder network: $\boldsymbol{\theta}_{E1},\ \boldsymbol{\theta}_{E2}$
4) Decoder network: $\boldsymbol{\theta}_{D1},\ \boldsymbol{\theta}_{D2}$

Step 1: Fit Propensity Networks

$\boldsymbol{\theta}_{A1} \leftarrow \texttt{optimize}\Big(\sum_{n,i} \texttt{binary\_x\_entropy}(r_{A1}(\mathbf{A}_n(i)|\bar{\mathbf{A}}_{n-1}(i); \boldsymbol{\theta}_{A1}), \mathbf{A}_n(i)) \Big)$

$\boldsymbol{\theta}_{A2} \leftarrow \texttt{optimize}\Big(\sum_{n,i} \texttt{binary\_x\_entropy}(r_{A2}(\mathbf{A}_n(i)|\bar{\mathbf{H}}_n(i); \boldsymbol{\theta}_{A2}), \mathbf{A}_n(i)) \Big)$

$\boldsymbol{\theta}_{C1} \leftarrow \texttt{optimize}\Big(\sum_{n,i} \texttt{binary\_x\_entropy}(r_{C1}(\bar{\mathbf{A}}_{\mathbf{n-1}}(\mathbf{i}); \boldsymbol{\theta}_{C1}(i)), \mathbf{C}_n(i)) \Big)$

$\boldsymbol{\theta}_{C2} \leftarrow \texttt{optimize}\Big(\sum_{n,i} \texttt{binary\_x\_entropy}(r_{C2}(\bar{\mathbf{L}}_{n-1}(i), \bar{\mathbf{A}}_{n-1}(i), \mathbf{X}(i); \boldsymbol{\theta}_{C2}), \mathbf{C}_n(i)) \Big)$

Step 2: Generate IPTWs
for patient $i = 1$ to I **do**
  for $t = 1$ to T **do**
    for $\tau = 1$ to τ_{max} **do**
      $\mathbf{SW}_i(t,\tau) \leftarrow \prod_{n=t}^{t+\tau} r_{A1}(\mathbf{A}_n(i)|\bar{\mathbf{A}}_{n-1}(i); \boldsymbol{\theta}_{A1}) \ / \ r_{A2}(\mathbf{A}_n(i)|\bar{\mathbf{H}}_n(i); \boldsymbol{\theta}_{A2})$
      $SW^*_i(t,\tau) \leftarrow \prod_{n=t}^{t+\tau} r_{C1}(\bar{\mathbf{A}}_{\mathbf{n-1}}(\mathbf{i}); \boldsymbol{\theta}_{C1}(i)) \ / \ r_{C2}(\bar{\mathbf{L}}_{n-1}(i), \bar{\mathbf{A}}_{n-1}(i), \mathbf{X}(i); \boldsymbol{\theta}_{C2})$
    end for
  end for
end for

Step 3: Fit Encoder
$\boldsymbol{\theta}_E \leftarrow \texttt{optimize}\big(\mathcal{L}_{encoder}\big)$, as per Equation 5a

Step 4: Compute Encoder States {Used to Initialize Decoder}
for patient $i = 1$ to I **do**
  for $t = 1$ to T **do**
    $\mathbf{h}_t(i) \leftarrow \tilde{g}_E(\bar{\mathbf{L}}_t(i), \bar{\mathbf{A}}_t(i), \mathbf{X}(i); \boldsymbol{\theta}_E)$
  end for
end for

Step 5: Fit Decoder
$\boldsymbol{\theta}_D \leftarrow \texttt{optimize}\big(\mathcal{L}_{decoder}\big)$, as per Equation 5b

F Hyperparameter Optimization for R-MSN

For the R-MSN, 10,000 simulated paths were used for backpropagation of the network (training data), and 1,000 simulated paths for hyperparameter optimization (validation data) - with another 1,000 for out-of-sample testing. Given the differences in state initialization requirements and data batching of the decoder, we report the hyperparameter optimization settings separately for the decoder. The optimal parameters of all networks can be found in Table 5.

Settings for Propensity Networks and Encoder Hyperparameter optimization was performed using 50 iterations of random search, using the hyperparameter ranges in Table 3, and networks were trained using the ADAM optimizer [20]. For each set of sampled, simulation trajectories were grouped into B minibatches and networks were trained for a maximum of 100 epochs. LSTM state sizes were also defined in relation to the number of inputs for the network C.

Table 3: Hyperparameter Search Range for Propensity Networks and Encoder

	Hyperparameter Search Range
Hyperparameter Search Iterations	50
Dropout Rate	0.1 , 0.2 , 0.3, 0.4, 0.5
State Size	0.5C, 1C, 2C, 3C, 4C
Minibatch Size	64, 128, 256
Learning Rate	0.01, 0.005, 0.001
Max Gradient Norm	0.5, 1.0, 2.0

Settings for Decoder To train the decoder, the data was reformatted into sequences of $(\mathbf{h}_t, \{\mathbf{L}_{t+1}, \ldots, \mathbf{L}_{t+\tau_{max}}\}, \{\mathbf{A}_t, \ldots, \mathbf{A}_{t+\tau_{max}}, \mathbf{X}\})$, such that each patient i max T_i contributions to the training dataset. Given the T-fold increase in the number of rows in the overall dataset, we made a few modifications to the range of hyperparameter search, including increasing the size of minibatches and reducing the learning rate and number of iterations of hyperparameter search. The full range of hyperparameter search can be found in Table 4 and networks are trained for maximum of 100 epochs as well.

Table 4: Hyperparameter Search Range for Decoder

	Hyperparameter Search Range
Iterations of Hyperparameter Search	20
Dropout Rate	0.1 , 0.2 , 0.3, 0.4, 0.5
State Size	1C, 2C, 4C, 8C, 16C
Minibatch Size	256, 512, 1024
Learning Rate	0.01, 0.005, 0.001, 0.0001
Max Gradient Norm	0.5, 1.0, 2.0, 4.0

Table 5: Optimal Hyperparameters for R-MSN

	Dropout Rate	**State Size**	**Minibatch Size**	**Learning Rate**	**Max Norm**
Propensity Networks					
$f(\mathbf{A}_n \vert \bar{\mathbf{A}}_{n-1})$	0.1	6 (3C)	128	0.01	2.0
$f(\mathbf{A}_n \vert \tilde{H}_n)$	0.1	16 (4C)	64	0.01	1.0
$f(C_n = 0 \vert \mathcal{T} > n, \bar{\mathbf{A}}_{\mathbf{n-1}})$	0.2	4 (2C)	128	0.01	0.5
$f(C_t = 0 \vert \mathcal{T} > n, \bar{\mathbf{L}}_{n-1}, \bar{\mathbf{A}}_{n-1}, \mathbf{X})$	0.1	16 (4C)	64	0.01	2.0
Prediction Networks					
Encoder	0.1	16 (4C)	64	0.01	0.5
Decoder + Memory Adapter	0.1	16 (8C)	512	0.001	4.0

G Hyperparameter Optimization for BTRC

The parameters of the BTRC were optimized using the maximum-a-posteriori (MAP) estimation, using the same prior for global parameters and approach defined in [35]. While the model was replicated as faithfully to the specifications as possible, two slight modifications were made to adapt it to our problem. Firstly, the sparse GP approximations were avoided to ensure that we had as much accuracy as possible - using Gaussian Process with full covariance matrices for the random effects components. Secondly, as our dataset was partitioned to ensure that patient observed in the training set were not present in the test set, this means that any patient-specific parameters learned would not be used in the testing set itself. As such, to avoid optimizing on the test set, we adopt the standard approach for prediction in generalized linear mixed models [25], using the average population parameters, i.e. the global MAP estimate, for prediction.

Hyperparameter optimization was performed using grid search on the optimizer settings defined in 6, and was performed for a maximum of 5000 epochs per configuration. As convergence was observed to be slow for a number of settings, we also trained a reduced form of the full BTRC model without the "shared" parameters (indicated by '-' in Table7) to reduce the number of parameters of the model. The optimal global hyperparameters and optimizer settings can be found in Table 7.

Table 6: Hyperparameter Grid for BTRC

	Hyperparameter Search Range
Minibatch Size	2, 5, 10, 100, 500
Learning Rate	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}

Table 7: MAP Estimates for BTRC

	BTRC	**R-BTRC**
$\bar{\chi}_{chemo}, \bar{\chi}_{radio}$	(-1.3729433, 0.065)	(-1.162, 0.007)
$\bar{\alpha}_{chemo}, \bar{\alpha}_{radio}$, $\bar{\alpha}^{(0)}_{chemo}, \bar{\alpha}^{(0)}_{radio}$	(0.760, 0.367), (0.207, 0.490)	(0.547, 0.367), (-, -)
$\bar{\beta}_{chemo}$,$\bar{\beta}_{radio}$, $\bar{\beta}^{(0)}_{chemo}, \bar{\beta}^{(0)}_{radio}$	(0.595, 0.367), (0.204, 0.368)	(0.429, 0.368), (-, -)
$\bar{\gamma}$	-0.27	-0.262
$\bar{\omega}$	-0.928	-
$\bar{l}^g$	1.223	-
$\bar{\kappa}$	0.786	0.867
$\bar{l}^v$	1.092	1.151
$\bar{\sigma}^2$	0.036	0.042
Learning Rate	0.001	0.001
Minibatch Size	100	100